WEST

Freeform Search

Database: US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Thursday, October 31, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L9	L6 and network$	10	L9
L8	L6 and e-mail	0	L8
L7	L6 and server$	0	L7
L6	L2 and ((decoder$ with extract$) same (interrupt$))	13	L6
L5	L2 and ((decoder$ with extract$) and (server$ and interrupt$))	5	L5
L4	L2 and ((decoder$ with extract$) and (server$ same interrupt$))	0	L4
L3	L2 and (decoder$ with interrupt$ with extract$)	0	L3
L2	(712/$.ccls.) or (710/$.ccls.)	19880	L2
L1	(712/$.ccls.) or (710/$.ccls.)	19880	L1

END OF SEARCH HISTORY

WEST

Generate Collection

L12: Entry 6 of 6 File: USPT Sep 21, 1999

DOCUMENT-IDENTIFIER: US 5956521 A
TITLE: System for universal electronic mail delivery where messaging devices are notified using a particular dialing, ringing, and hanging-up pattern

Detailed Description Text (44):
Referring to Appendix A, the software pseudo-code for the client's e-mail device is illustrated. When the device is first turned on, a power-on self-test is executed. If there is a fatal failure, the program flow branches to the Fatal.sub.-- Error.sub.-- Stop label, sets the fatal error indicator, and halts the system. If a minor failure occurred, the program flow branches to the Warning.sub.-- Code label, sets a warning code indicator and resumes the program flow. Next, the phone line status is checked. If it is busy, the device will wait until the line is not busy. The e-mail device is then placed in auto-answer mode and the registers for the device are initialized for operation. If there is any failure during this initialization process, a warning code is posted. After the initialization process, the software continuously loops to check for an interrupt from the interrupt registers. If an interrupt is found, the program branches to the Interrupt.sub.-- Service routine. The Interrupt.sub.-- Service routine reads the interrupt register, determines the interrupt type, and branches to the corresponding interrupt routine.

Detailed Description Text (50):
If the Registration.sub.-- Request interrupt flag is set, this flag indicates that the client has placed the device in registration mode in order to register with the main server. This process is generally executed when the device is being set up for the first time or when the device has been moved to a new location. The program flow branches to the Registration.sub.-- Request routine, where the device dials a designated phone number for registration. Generally, this is a 800 toll free number connected to the main server. When connected, the device delivers the machine ID, the security code, and the client's phone number to the main server. The main server determines the particular local server for serving the client's e-mail device based upon the given phone number. The phone number for the particular local server is sent to the client device, and the client device retains the number in memory for later use.

Detailed Description Text (57):
In another hardware embodiment, the e-mail device is an integral part of a computer expansion card having power supplied from two sources, the computer system itself or an external power supply. Referring to FIG. 8a, an expansion card 50 having an edge connector 52 is illustrated. The expansion card is insertable into an edge connector slot connected to the bus of a computer system. The expansion card includes a CPU 54 (or microcontroller) directly polling an I/O register 56 that is communicatively connected to a notification module 58. The I/O register 56 receives information from the notification module 58 and the user input and control device 57 (which can be a keyboard, a keypad, dip switches, etc.) for entering security code, e-mail messages, or other inputs, and generates signals for indicators 59 to indicate the status of any messages and the e-mail device. The notification module sends and receives information via a phone line connection and interacts with the communication module 62. When the expansion card is inserted into the computer system, a bus controller 64 controls the data flow to and from the computer system (not shown) via the edge connectors 52. Information is passed between the flash memory 66, the ROM 68, the RAM 70, the CPU 54, and the communication module 62 through an internal bus 72. The communication module can be a fax/modem chipset. The expansion card 50 may be powered by one of two sources, power from the computer system via trace 74 or power from an external source via trace 76 and power jack 78. The power switching and conversion module 80 detects power from one of the two sources, performs any power

conversion from one voltage level to another voltage level if it is needed, and routes the power to the components on the expansion card 50. The power detection and switching is automatically performed without interruption to the operation of the e-mail device. Thus, no interruption of operation would occur if power is switched in the midst of sending or receiving e-mail messages.

Current US Original Classification (1):
710/35

Current US Cross Reference Classification (1):
710/3

Current US Cross Reference Classification (2):
710/33

US005956521A

United States Patent [19]

Wang

[11] Patent Number: 5,956,521

[45] Date of Patent: Sep. 21, 1999

[54] **SYSTEM FOR UNIVERSAL ELECTRONIC MAIL DELIVERY WHERE MESSAGING DEVICES ARE NOTIFIED USING A PARTICULAR DIALING, RINGING, AND HANGING-UP PATTERN**

[76] Inventor: **Kevin Kuan-Pin Wang**, 11867 Woodhill Ct., Cupertino, Calif. 95014

[21] Appl. No.: **08/656,651**

[22] Filed: **May 31, 1996**

Related U.S. Application Data

[63] Continuation-in-part of application No. 08/494,652, Jun. 26, 1995, Pat. No. 5,757,891.

[51] **Int. Cl.**[6] **G06F 15/16**

[52] **U.S. Cl.** **395/855**; 395/823; 395/853

[58] **Field of Search** 395/200.32, 200.36, 395/200.33, 200.37, 200.45, 823, 853, 855; 379/88, 93.05, 198, 93.24, 100.09, 113, 265

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,451,701	5/1984	Bendig	379/93.25
4,630,196	12/1986	Bednar, Jr. et al.	395/200.32
4,757,267	7/1988	Riskin	379/113
4,837,797	6/1989	Freeny, Jr.	379/93.17
4,924,496	5/1990	Figa et al.	379/142
5,003,580	3/1991	Duong et al.	379/93.09
5,014,300	5/1991	Harvath et al.	379/100.09
5,127,087	6/1992	Kasiraj et al.	395/200.36
5,130,977	7/1992	May et al.	370/422
5,138,653	8/1992	Le Clereq	379/93.24
5,163,156	11/1992	Leung et al.	395/200.45
5,164,982	11/1992	Davis	379/93.17
5,245,651	9/1993	Takashima et al.	379/93.23
5,293,250	3/1994	Okumura et al.	358/402
5,333,152	7/1994	Wilber	379/102.04
5,379,340	1/1995	Overend et al.	379/93.24
5,426,594	6/1995	Wright et al.	395/200.36
5,467,385	11/1995	Reuben et al.	379/88
5,483,466	1/1996	Kawahara et al.	395/200.33
5,530,744	6/1996	Charalambous et al.	379/265
5,535,407	7/1996	Yanagawa et al.	705/39
5,555,100	9/1996	Bloomfield et al.	358/402
5,596,573	1/1997	Bertland	370/474
5,627,764	5/1997	Schutzman et al.	395/200.37
5,684,862	11/1997	Finnigan	379/88
5,727,047	3/1998	Bently et al.	379/93.05
5,734,901	3/1998	Sidhu et al.	395/680
5,737,400	4/1998	Bagchi et al.	379/198
5,742,905	4/1998	Pepe et al.	455/461
5,754,778	5/1998	Shoujima	395/200.36
5,799,071	8/1998	Azar et al.	379/113

OTHER PUBLICATIONS

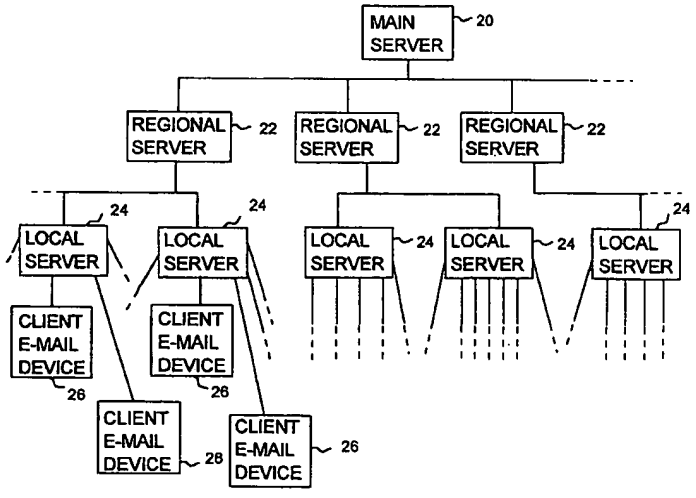
Krol, Ed, "The Whole Internet User'Guide & Catalog", O'Reilly & Associates, Inc., May 1993, pp. 19–30, 91–100 and 115–126.

Primary Examiner—Thomas C. Lee
Assistant Examiner—Chien Yuan
Attorney, Agent, or Firm—Claude A. S. Hamrick; Oppenheimer W. Donnelly; Emil Chang

[57] **ABSTRACT**

A system for facilitating, sending and receiving e-mail messages is disclosed. This e-mail system is supported by one or more main servers and a plurality of regional servers geographically distributed in populated areas, and are interconnected via a computer network such as the internet. An incoming e-mail message under this system is first processed and packaged by the main server to allow tracking of this message. The packaged message is then sent to the designated local server via a regional server. The local server receives the e-mail message and notifies or delivers the message to a client (user) e-mail device through one of several available notification methods. The e-mail device is a novel device designed to send and receive e-mail messages. It is a low cost device that may be a stand-alone device, a part of a multi-function device, or a part of a computer expansion card. The servers of the present invention can be maintained and operated remotely.

21 Claims, 18 Drawing Sheets



Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L12	l1 and ((e-mail$) same (interrupt$))	6	L12
L11	l1 and ((e-mail$) with (interrupt$))	1	L11
L10	l7 and (network$)	0	L10
L9	L1 and (priority with (exception$ or interrupt$) with handler$) and (network$ and e-mail$)	2	L9
L8	L1 and (priority with (exception$ or interrupt$) with handler$) and (network$)	72	L8
L7	L1 and (priority with (exception$ or interrupt$) with handler$) and (extract$ with field$ with decod$)	1	L7
L6	L1 and (priority with (exception$ or interrupt$) with handler$) and (extract$)	51	L6
L5	L1 and (priority with (exception$ or interrupt$) with handler$) and (extractor$)	0	L5
L4	L1 and (priority with (exception$ or interrupt$) with handler$) same (extractor$)	0	L4
L3	L1 and (priority with (exception$ or interrupt$) with handler$)	147	L3
L2	L1 and (priority with (exception$ or interrupt$))	2113	L2
L1	(712/$.ccls.) or (710/$.ccls.)	19880	L1

END OF SEARCH HISTORY

decoder interrupt
extract

WEST

Generate Collection

L7: Entry 1 of 1 File: USPT Jul 17, 2001

DOCUMENT-IDENTIFIER: US 6263396 B1
TITLE: Programmable interrupt controller with interrupt set/reset register and dynamically alterable interrupt mask for a single interrupt processor

Abstract Text (1):
A programmable interrupt controller (510) for a single interrupt architecture processor (518) includes a plurality of interrupt sources (502) each operable to generate an interrupt. A dynamically alterable interrupt mask (508) selectively blocks interrupt signals for the interrupt sources (502). Interrupts permitted by the dynamically alterable interrupt mask (508) are processed by an interrupt handler (500) for the single interrupt architecture processor (518) in order of priority. In addition, processing for a lower priority interrupt is interrupted in order to process a later received higher priority interrupt permitted by the dynamically alterable interrupt mask (508).

Brief Summary Text (30):
In accordance with one embodiment of the present invention, a programmable interrupt controller for a single interrupt architecture processor includes a plurality of interrupt sources each operable to generate an interrupt. A dynamically alterable interrupt mask selectively blocks interrupt signals for the interrupt sources. Interrupts permitted by the dynamically alterable interrupt mask are processed by an interrupt handler for the single interrupt architecture processor in order of priority. In addition, processing for a lower priority interrupt is interrupted in order to process a later received higher priority interrupt permitted by the dynamically alterable interrupt mask.

Detailed Description Text (197):
Besides normal bitstream decoding, the video decoder 252 also extracts from the picture layer user data the Closed Caption (CC), the Extended Data Services (EDS), the Presentation Time Stamps (PTS) and Decode Time Stamps, the pan_and_scan, the fields display flags, and the no_burst flag. These data fields are specified by the DSS. The CC and EDS are forwarded to the NTSC/PAL encoder module 260 and the PTS is used for presentation synchronization. The other data fields form a DSS-specific constraints on the normal MPEG bitstream, and they are used to update information obtained from the bitstream.

Detailed Description Text (448):
At reset, the FIQ interrupt logic is in an undetermined state. Initially, the FIQ handler 500 will reprogram all interrupt sources 502 as unmasked to enable all modules to interrupt with all logical interrupts at equal priority. Each interrupt will be reset to unpend any that may have occurred before a warm reset.

Current US Original Classification (1):
710/263

Current US Cross Reference Classification (1):
710/266

Current US Cross Reference Classification (2):
710/268

CLAIMS:

1. A programmable interrupt controller for a single interrupt architecture processor, comprising:

a plurality of interrupt sources each operable to generate an interrupt;

a dynamically alterable interrupt mask operable to selectively mask interrupts for the interrupt sources;

an interrupt handler for the single interrupt architecture processor, the interrupt handler operable to process interrupts permitted by the interrupt mask in order of priority and to interrupt processing for a lower priority interrupt in order to process a later-received higher priority interrupt permitted by the dynamically alterable interrupt mask;

an interrupt set/reset register disposed between the interrupt sources and the dynamically alterable interrupt mask, the interrupt set/reset register operable to store an interrupt for each of the interrupt sources;

wherein the interrupt sources include hardware and firmware devices and the firmware devices are operable to control interrupts for the hardware devices in the set/reset register; and

wherein the interrupt handler is further operable to save processing context for the lower priority interrupt prior to processing the higher priority interrupt and to restore processing context for further processing of the lower priority interrupt after completion of processing for the higher priority interrupt.

US006263396B1

(12) **United States Patent**
Cottle et al.

(10) **Patent No.: US 6,263,396 B1**
(45) **Date of Patent: Jul. 17, 2001**

(54) **PROGRAMMABLE INTERRUPT CONTROLLER WITH INTERRUPT SET/RESET REGISTER AND DYNAMICALLY ALTERABLE INTERRUPT MASK FOR A SINGLE INTERRUPT PROCESSOR**

(75) Inventors: **Temple D. Cottle**, Hurst; **Tiemen T. Spits**, North Richland Hills, both of TX (US)

(73) Assignee: **Texas Instruments Incorporated**, Dallas, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/235,665**

(22) Filed: **Jan. 22, 1999**

Related U.S. Application Data

(63) Continuation of application No. 08/962,514, filed on Oct. 31, 1997.

(60) Provisional application No. 60/030,107, filed on Nov. 1, 1996, provisional application No. 60/030,106, filed on Nov. 1, 1996, provisional application No. 60/030,105, filed on Nov. 1, 1996, provisional application No. 60/030,104, filed on Nov. 1, 1996, provisional application No. 60/030,108, filed on Nov. 1, 1996, and provisional application No. 60/029,923, filed on Nov. 1, 1996.

(51) **Int. Cl.**[7] **G06F 13/24**; G06F 13/26

(52) **U.S. Cl.** **710/263**; 710/266; 710/268

(58) **Field of Search** 710/261, 262, 710/264, 265, 266, 268, 269

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,083,261	*	1/1992	Wilkie	710/266
5,425,061	*	6/1995	Laczko, Sr. et al.	375/371
5,594,905		1/1997	Mital	395/733
5,603,035		2/1997	Erramoun et al.	395/733
5,608,459	*	3/1997	Hashimoto et al.	348/416
5,729,556	*	3/1998	Benbassat et al.	371/31
5,758,168		5/1998	Mealey et al.	395/733
5,845,239	*	12/1998	Laczko, Sr. et al.	704/200

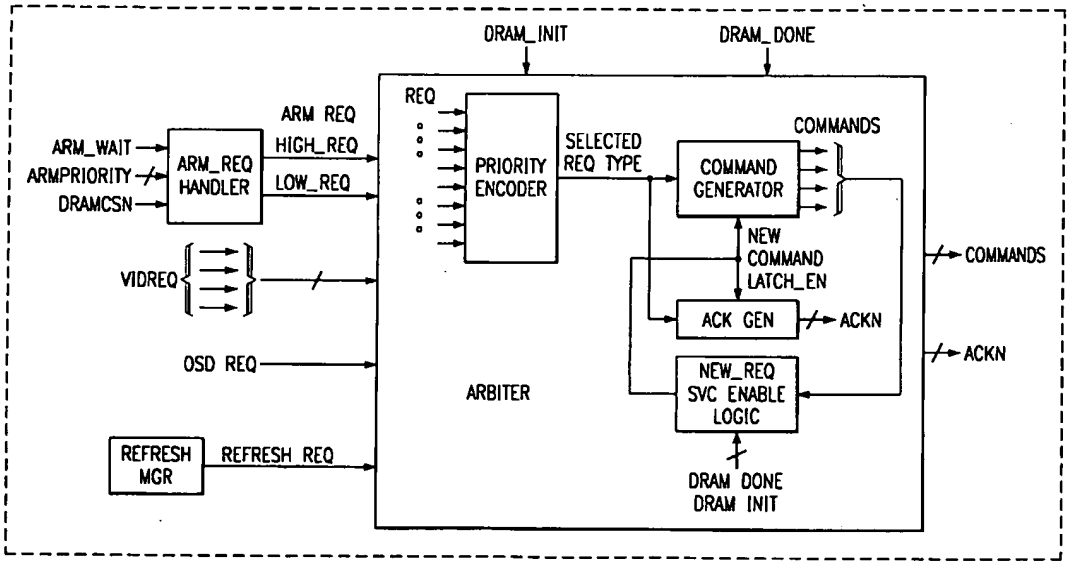
* cited by examiner

Primary Examiner—Robert Beausoleil
Assistant Examiner—Raymond N Phan
(74) *Attorney, Agent, or Firm*—Robert D. Marshall, Jr.; W. James Brady, III; Frederick J. Telecky, Jr.

(57) **ABSTRACT**

A programmable interrupt controller (**510**) for a single interrupt architecture processor (**518**) includes a plurality of interrupt sources (**502**) each operable to generate an interrupt. A dynamically alterable interrupt mask (**508**) selectively blocks interrupt signals for the interrupt sources (**502**). Interrupts permitted by the dynamically alterable interrupt mask (**508**) are processed by an interrupt handler (**500**) for the single interrupt architecture processor (**518**) in order of priority. In addition, processing for a lower priority interrupt is interrupted in order to process a later received higher priority interrupt permitted by the dynamically alterable interrupt mask (**508**).

9 Claims, 76 Drawing Sheets



L12: Entry 1 of 6 File: USPT Sep 3, 2002

DOCUMENT-IDENTIFIER: US 6446192 B1
TITLE: Remote monitoring and control of equipment over computer networks using a single web interfacing chip

Detailed Description Text (30):
The Network Peripheral with API mode is considerably more sophisticated. For systems needing more functionality than pass-through mode offers, the network interface chip can act as a slave peripheral with a command based API (Application Programming Interface). As a peripheral, the network interface chip's API gives the equipment access to services such as e-mail, Sockets, and a monitoring and control protocol. These services enable the device to give Web access to the system's internal variables and functions, send e-mail notification, and can even manage multiple clients. As shown in FIG. 2, the network interface chip has a variety of modules in its device interface, implementing an API with many commands to accommodate a wide variety of applications and methods of network interfacing. All of these commands use a common protocol that is simple and efficient, requiring minimal programming and processor overhead to support. Although there are many commands in the API, a given application will use a small subset of the available commands, depending on the style of interface and communication chosen by the designer. The commands are divided into groups according to the service classes. For a given application, the designer will typically choose one of the following methods of network access and use only the class of commands associated with that method: Monitoring and Control Interface: This API subset provides the remote client the ability to query and change values (i.e., variables) in the device. E-mail Interface: This API implements E-mail transmission using SMTP. E-mail messages may be used to notify recipients of events in the device. Sockets Interface: The sockets API implements a BSD Sockets type interface. This sets up a full-duplex data stream between the remote client and the device. It provides ultimate flexibility for implementing more sophisticated client or server applications and moving large amounts of data. Datagram Interface: This is a simple packet-based interface using the UDP transport. It allows packets of data to be sent to and received from remote clients. Remote Procedure Call Interface: This API subset provides the remote client the ability to make procedure calls (i.e., execute functions) in the system. FTP: File Transfer Protocol is the Internet standard for downloading and uploading files between devices. Files are stored in the host equipment's file system. The network interface chip handles all the FTP server functionality, interrupting the host processor only to read or write a file. HTML Page Mode Interface: The network interface chip currently can act as a Web server, serving pages stored in an external EEPROM. HTML pages may be stored in the main memory of the host equipment's processor. The network interface chip will request a specific page, and the equipment processor will retrieve the page from memory and write it to the network interface chip.

Detailed Description Text (44):
In Parallel Peripheral Mode, the network interface chip is a slave peripheral and implements a command API over the bi-directional processor bus. The device processor initiates all transactions, although the network interface chip may request attention by asynchronously generating an interrupt. In polled systems, attention is requested through the status register which must be read periodically by the processor. The network interface chip API has many commands to accommodate a wide variety of applications and methods of network interfacing. All of these commands use a common protocol that is simple to understand and use, and requires minimal programming and processor overhead to support. The API is described later in this document. The programmable I/O device interface will now be described in relation to FIG. 11. For systems without a processor, a number N of general purpose digital I/O lines are available for sensing and control. In a preferred embodiment, N=24. Each

I/O of the N lines may be configured for input, output, or both. Inputs can be used for sensing switches, level detectors, or other status signals in the device. Lines configured as outputs can be used to drive relays, actuators, indicators, or other components. The remote client 30 has access to these I/O lines over the network 32, as described above. In addition to simple inputs and outputs, some I/O lines may be configured as edge-triggered inputs. Using edge-triggered configuration, an event can be captured until the remote client reads and acknowledges the event. Alternatively, an event may used to send a notification signal or an e-mail message. Each I/O line may be addressed individually, or multiple I/Os may: be grouped and controlled simultaneously as a bus. Although intended for devices without a processor, this interface could also be used with devices that do have a processor, but where there is no need for the network interface chip to interface directly with the processor.

Current US Original Classification (1):
712/29

Current US Cross Reference Classification (6):
712/28

Current US Cross Reference Classification (7):
712/31

US006446192B1

(12) **United States Patent**
Narasimhan et al.

(10) **Patent No.: US 6,446,192 B1**
(45) **Date of Patent: Sep. 3, 2002**

(54) **REMOTE MONITORING AND CONTROL OF EQUIPMENT OVER COMPUTER NETWORKS USING A SINGLE WEB INTERFACING CHIP**

(75) Inventors: **Subram Narasimhan**, Saratoga; **Curtis Allred**, Cupertino; **Mark Stemm**, Berkeley, all of CA (US); **Hari Balakrishnan**, Winchester, MA (US)

(73) Assignee: **Embrace Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/326,105**

(22) Filed: **Jun. 4, 1999**

(51) **Int. Cl.**[7] **G06F 9/54**; G06F 6/45; G06F 13/42; G06F 15/173

(52) **U.S. Cl.** **712/29**; 712/28; 712/31; 709/203; 709/230; 709/236; 709/211; 709/328; 717/118; 717/136; 717/148

(58) **Field of Search** 709/319, 320, 709/311, 332, 219, 210, 315, 329, 218, 203, 230, 224, 232, 226, 234, 236, 245, 250, 253, 328, 324, 211, 221, 209, 231; 235/462.15, 472.01; 712/217, 28, 30, 31, 29, 38, 32; 713/201, 167, 153, 165, 200, 151, 202; 702/122; 707/513, 501, 2, 103, 104, 5, 3, 10; 705/51.14, 54; 717/118, 136, 148

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,761,430 A * 6/1998 Gross et al. 709/225
5,790,977 A 8/1998 Ezekiel 702/122
5,905,248 A * 5/1999 Russell et al. 709/218
6,154,843 A * 11/2000 Hart, Jr. et al. 713/201

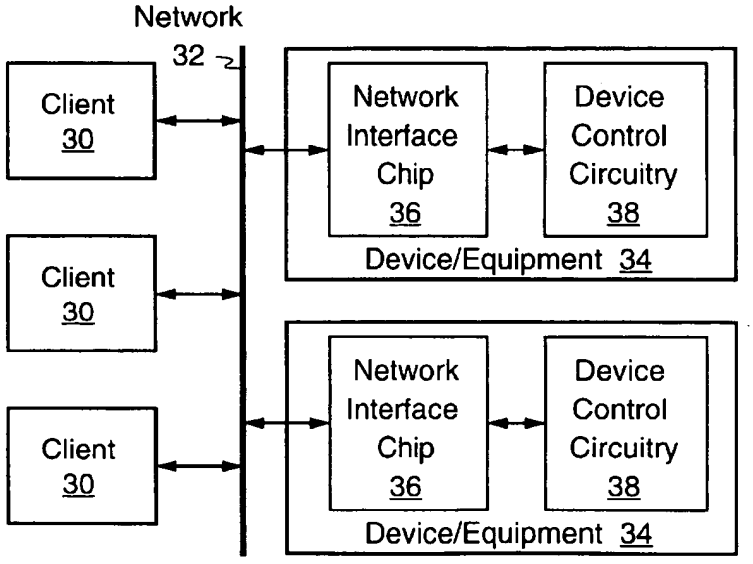
* cited by examiner

Primary Examiner—Daniel H. Pan
(74) *Attorney, Agent, or Firm*—Lumen Intellectual Property Services, Inc.

(57) **ABSTRACT**

A single integrated circuit chip interfaces device control circuitry of a device to a client machine via a computer network. The chip comprises an internal data bus; a central processing unit connected to the internal data bus; an internal memory connected to the internal data bus; a device interface connected to the internal data bus, wherein the device interface comprises circuit blocks for communicating digital information between the integrated circuit and the device control circuitry; and a network interface connected to the internal data bus, wherein the network interface comprises circuit blocks for communicating digital information between the integrated circuit and the computer network. The internal memory comprises instructions for implementing complete internet protocol functionality on the network; translating information between network protocol formats and a format of the device; transferring information between the network and the device control circuitry; and sending customized software to the client machine over the network, wherein the software is executable on the client machine, and wherein the software enables the client machine to generate device control signals and to receive device status information. The single integrated circuit chip provides inexpensive, compact, powerful, and versatile interfacing of a large variety of devices to high performance computer networks.

11 Claims, 12 Drawing Sheets



L12: Entry 5 of 6 File: USPT Feb 1, 2000

DOCUMENT-IDENTIFIER: US 6021429 A
TITLE: Network device which maintains a list of device addresses

Detailed Description Text (21):
The TCP/IP protocol stack has User Datagram Protocol (UDP), Reverse Address Resolution Protocol (RARP) and BootP support within. INTERRUPT is the interrupt handler for the TCP/IP task. LPRINTSERVER is the TCP/IP print server application, and also owns the print data lines for the duration of a print job. The TCP/IP protocol stack contains process steps of the present invention for determining whether a list manager, i.e., a network device which maintains a list of device addresses for LAN 1, is operating on LAN 1, controlling NEB 2 to provide its device address to a list manager when it is determined that a list manager is operating on the LAN, and controlling NEB 2 to operate as the list manager for the LAN when it is determined that no list manager is operating on the LAN. These device addresses can include, but are not limited to, a MAC address of NEB 2, or an E-Mail address of a PC, such as PC 26, in a case that the invention is connected with workstations on LAN 1. Each of the device addresses includes a socket number specific to the present invention. In this regard, an example of a device address which could be in a list maintained by the list manager is "146.184.24.51:25", where "25" is the socket number for the present invention.

Current US Cross Reference Classification (3):
710/9

US006021429A

United States Patent [19]

Danknick

[11] Patent Number: 6,021,429

[45] Date of Patent: Feb. 1, 2000

[54] **NETWORK DEVICE WHICH MAINTAINS A LIST OF DEVICE ADDRESSES**

[75] Inventor: **Dan Danknick**, Orange, Calif.

[73] Assignee: **Canon Information Systems, Inc.**, Irvine, Calif.

[21] Appl. No.: **08/749,629**

[22] Filed: **Nov. 18, 1996**

[51] **Int. Cl.**[7] **G06F 12/06**; G06F 13/38

[52] **U.S. Cl.** **709/208**; 710/9; 709/250; 709/245

[58] **Field of Search** 395/200.53, 200.54, 395/200.38, 200.39, 200.48, 200.67, 200.75, 200.79, 200.8, 200.83, 823, 824, 828, 829, 835, 840, 200.52; 709/218, 219, 208, 209, 223, 224, 247, 249, 250, 253; 710/3, 4, 8, 9

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,140,585	8/1992	Tomikawa	370/60.1
5,185,860	2/1993	Wu	395/200
5,287,103	2/1994	Kasprzyk et al.	340/825.52
5,304,992	4/1994	Harashima	340/825.52
5,337,309	8/1994	Faulk, Jr.	370/60
5,353,399	10/1994	Kuwamoto et al.	395/159
5,542,047	7/1996	Armstrong	395/200.11
5,548,722	8/1996	Jalalian	395/200.1
5,548,725	8/1996	Tanaka et al.	395/200.05
5,550,979	8/1996	Tanaka et al.	395/200.05
5,588,119	12/1996	Vincent et al.	395/200.15
5,604,869	2/1997	Mincher et al.	395/200.2
5,615,389	3/1997	Mayfield et al.	395/828
5,668,952	9/1997	Slane	395/200.75
5,687,320	11/1997	Wiley et al.	395/200.16
5,706,210	1/1998	Kumano et al.	364/514
5,727,157	3/1998	Orr et al.	395/200.54
5,751,967	5/1998	Raab et al.	395/200.58
5,754,767	5/1998	Ruiz	395/200.5
5,774,667	6/1998	Garvey et al.	395/200.52
5,802,300	9/1998	Tanaka et al.	395/200.52
5,838,907	11/1998	Hansen	395/200.5
5,845,081	12/1998	Rangarajan et al.	395/200.54

OTHER PUBLICATIONS

"HP Forms Internet Solutions Operation", Hewlett–Packard Printing & Imaging News (visited Nov. 6, 1996) <http://www–dmo.external.hp.com/peripherals/press/news/96jul15.html> (2 pages).

HP Unveils Strategy for Future of Printer Management, Hewlett–Packard Company (1995) (3 pages).

Tektronix PhaserLink Software: Web–Powered Print Administration and Support, Tektronix, Inc., Jan. 8, 1996 (1 page).

"PhaserLink Software: East access to your printer's information", Tektronix, Inc., Jan. 8, 1996, (7 pages).

"PhaserLink Demo", Tektronix, Inc. (visited Nov. 6, 1996) <http:/www.tek.com/Color_Printers/support/demo.html> (3 pages).

"PhaserLink for the Phaser 340: Current status for printer named Monterey/Wasatch", Tektronix, Inc. (visited Nov. 6, 1996) <http://www.tek.com/Color_Printers/support/demo340/button_status.html> (1 page).

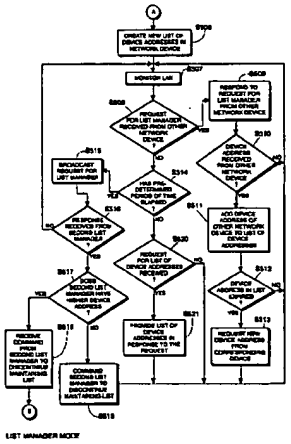
(List continued on next page.)

Primary Examiner—Frank J. Asta
Assistant Examiner—Marc D. Thompson
Attorney, Agent, or Firm—Fitzpatrick, Cella, Harper & Scinto

[57] **ABSTRACT**

A method controls a network device on a local area network (LAN) to operate as a list manager which maintains a list of device addresses for the LAN, and to operate as a slave which provides a device address of the network device to a list manager. The method comprises the steps of determining whether a list manager is operating on the LAN, controlling the network device to operate as a slave on the LAN when the determining step determines that a list manager is operating on the LAN, and controlling the network device to operate as the list manager for the LAN when the determining step determines that no list manager is operating on the LAN.

54 Claims, 10 Drawing Sheets



L9: Entry 1 of 2 File: USPT Aug 11, 1998

DOCUMENT-IDENTIFIER: US 5794062 A
TITLE: System and method for dynamically reconfigurable computing using a processing unit having changeable internal hardware organization

Abstract Text (1):
A set of S-machines, a T-machine corresponding to each S-machine, a General Purpose Interconnect Matrix (GPIM), a set of I/O T-machines, a set of I/O devices, and a master time-base unit form a system for scalable, parallel, dynamically reconfigurable computing. Each S-machine is a dynamically reconfigurable computer having a memory, a first local time-base unit, and a Dynamically Reconfigurable Processing Unit (DRPU). The DRPU is implemented using a reprogrammable logic device configured as an Instruction Fetch Unit (IFU), a Data Operate Unit (DOU), and an Address Operate Unit (AOU), each of which are selectively reconfigured during program execution in response to a reconfiguration interrupt or the selection of a reconfiguration directive embedded within a set of program instructions. Each reconfiguration interrupt and each reconfiguration directive references a configuration data set specifying a DRPU hardware organization optimized for the implementation of a particular Instruction Set Architecture (ISA). The IFU directs reconfiguration operations, instruction fetch and decode operations, memory access operations, and issues control signals to the DOU and the AOU to facilitate instruction execution. The DOU performs data computations, and the AOU performs address computations. Each T-machine is a data transfer device having a common interface and control unit, one or more interconnect I/O units, and a second local time-base unit. The GPIM is a scalable interconnect network that facilitates parallel communication between T-machines. The set of T-machines and the GPIM facilitate parallel communication between S-machines.

Brief Summary Text (40):
The GPIM provides a scalable point-to-point interconnect means for parallel communication between T-machines. The set of T-machines and the GPIM together form a scalable point-to-point interconnect means for parallel communication between S-machines. The GPIM preferably comprises a k-ary n-cube static interconnect network having a plurality of first communication channels and a plurality of second communication channels. Each first communication channel includes a plurality of node connection sites, as does each second communication channel. Each interconnect I/O unit in the system is coupled to the GPIM such that its input is coupled to a particular node connection site via a message input line, and its output is coupled to another node connection site via message output line. The GPIM is thus a scalable network for routing data and commands between multiple interconnect I/O units in parallel.

Detailed Description Text (34):
The interrupt logic 106 preferably comprises a state machine that generates transition control signals, and performs interrupt notification operations in response to an interrupt signal received via the external control line 48. Referring now to FIG. 7, a state diagram showing a preferred set of states supported by the interrupt logic 106 is shown. The interrupt logic 106 begins operation in state P. State P corresponds to a power-on, reset, or reconfiguration condition. In response to the completion signal issued by the reconfiguration logic 104, the interrupt logic 106 advances to state A and retrieves the interrupt response signals from the architecture description memory 101. The interrupt logic 106 then generates the transition control signals from the interrupt response signals, and stores the transition control signals in the process control register set 122. In the preferred embodiment, the interrupt logic 106 includes a CLB-based Programmable Logic Array (PLA) for receiving the interrupt response signals and generating the transition

control signals. Following state A, the interrupt logic 106 advances to state B to wait for an interrupt signal. Upon receipt of an interrupt signal, the interrupt logic 106 advances to state C in the event that the interrupt masking flag within the process control register set 122 is reset. Once in state C, the interrupt logic 106 determines the origin of the interrupt, an interrupt priority, and an interrupt handler address. In the event that the interrupt signal is a reconfiguration interrupt, the interrupt logic 106 advances to state R and stores a configuration data set address in the process control register set 122. After state R, or following state C in the event that the interrupt signal is not a reconfiguration interrupt, the interrupt logic 106 advances to state N and stores the interrupt handler address in the process control register set 122. The interrupt logic 106 next advances to state X, and issues an interrupt notification signal to the ISS 100. Following state X, the interrupt logic 122 returns to state B to wait for a next interrupt signal.

Detailed Description Text (44):
The data operate logic 154 performs arithmetic, shifting, and/or logical operations upon data received at its data input in response to the DOU control signals received at its control input. The store/align logic 152 comprises data storage elements that provide temporary storage for operands, constants, and partial results associated with data computations, under the direction of RF addresses and DOU control signals received at its address input and control input, respectively. The DOU cross-bar switch 150 is preferably a conventional cross-bar switch network that facilitates the loading of data from the memory 34, the transfer of results output by the data operate logic 154 to the store/align logic 152 or the memory 34, and the loading of constants output by the IFU 60 into the store/align logic 152 in accordance with the DOU control signals received at its control input. In the preferred embodiment, the detailed structure of the data operate logic 154 is dependent upon the types of operations supported by the ISA currently under consideration. That is, the data operate logic 154 comprises circuitry for performing the arithmetic and/or logical operations specified by the data-operate instructions within the currently-considered ISA. Similarly, the detailed structure of the store/align logic 152 and the DOU cross-bar switch 150 is dependent upon the ISA currently under consideration. The detailed structure of the data operate logic 154, the store/align logic 152, and the DOU cross-bar switch 150 according to ISA type is described hereafter with reference to FIGS. 9A and 9B.

Detailed Description Text (45):
For an outer-loop ISA, the DOU 62 is preferably configured to perform serial operations upon data. Referring now to FIG. 9A, a block diagram of a first exemplary embodiment of the DOU 61 configured for the implementation of a general-purpose outer-loop ISA is shown. A general-purpose outer-loop ISA requires hardware configured for performing mathematical operations such as multiplication, addition, and subtraction; Boolean operations such as AND, OR, and NOT; shifting operations; and rotating operations. Thus, for the implementation of a general-purpose outer-loop ISA, the data operate logic 154 preferably comprises a conventional Arithmetic-Logic Unit (ALU)/shifter 184 having a first input, a second input, a control input, and an output. The Store/Align logic 152 preferably comprises a first RAM 180 and a second RAM 182, each of which has a data input, a data output, an address-select input, and an enable input. The DOU cross-bar switch 150 preferably comprises a conventional cross-bar switch network having both bidirectional and unidirectional crossbar couplings, and having the inputs and outputs previously described with reference to FIG. 8. Those skilled in the art will recognize that an efficient implementation of the DOU cross-bar switch 150 for an outer-loop ISA may include multiplexors, tri-state buffers, CLB-based logic, direct wiring, or subsets of the aforementioned elements joined in any combination by virtue of reconfigurable coupling means. For an outer-loop ISA, the DOU cross-bar switch 150 is implemented to expedite serial data movement in a minimum possible time, while also providing a maximum number of unique data movement cross-bar couplings to support generalized outer-loop instruction types.

Detailed Description Text (48):
Referring now to FIG. 9B, a block diagram of a second exemplary embodiment of the DOU 63 configured for the implementation of an inner-loop ISA is shown. In general, an inner-loop ISA supports relatively few, specialized operations, and is preferably

used to perform a common set of operations upon potentially large data sets. Optimum computational performance for an inner-loop ISA is therefore produced by hardware configured to perform operations in parallel. Thus, in the second exemplary embodiment of the DOU 63, the data operate logic 154, the store/align logic 152, and the DOU cross-bar switch 150 are configured to perform pipelined computations. The data operate logic 154 comprises a pipelined functional unit 194 having a plurality of inputs, a control input, and an output. The store/align logic 152 comprises: 1) a set of conventional flip-flop arrays 192, each flip-flop array 192 having a data input, a data output, and a control input; and 2) a data selector 190 having a control input, a data input, and a number of data outputs corresponding to the number of flip-flop arrays 192 present. The DOU cross-bar switch 150 comprises a conventional cross-bar switch network having duplex unidirectional crossbar couplings. In the second exemplary embodiment of the DOU 63, the DOU cross-bar switch 150 preferably includes the inputs and outputs previously described with reference to FIG. 8, with the exception of the second data feedback input. In a manner analogous to the outer-loop ISA case, an efficient implementation of the DOU cross-bar switch 150 for an inner-loop ISA may include multiplexors, tri-state buffers, CLB-based logic, direct wiring, or a subset of the aforementioned elements coupled in a reconfigurable manner. For an inner-loop ISA, the DOU cross-bar switch 150 is preferably implemented to maximize parallel data movement in a minimum amount of time, while also providing a minimum number of unique data movement cross-bar couplings to support heavily pipelined inner-loop ISA instructions.

Detailed Description Text (53):
Referring now to FIG. 11A, a block diagram of a first exemplary embodiment of the AOU 65 configured for the implementation of a generalpurpose outer-loop ISA is shown. A general-purpose outer-loop ISA requires hardware for performing operations such as addition, subtraction, increment, and decrement upon the contents of a program counter and addresses stored in the store/count logic 202. In the first exemplary embodiment of the AOU 65, the address operate logic 204 preferably comprises a Next Instruction Program Address Register (NIPAR) 232 having an input, an output, and a control input; an arithmetic unit 234 having a first input, a second input, a third input, a control input, and an output; and a multiplexor 230 having a first input, a second input, a control input, and an output. The store/count logic 202 preferably comprises a third RAM 220 and a fourth RAM 222, each of which has an input, an output, an address-select input, and an enable input. The address multiplexor 206 preferably comprises a multiplexor having a first input, a second input, a third input, a control input, and an output. The AOU cross-bar switch 200 preferably comprises a conventional cross-bar switch network having duplex unidirectional crossbar couplings, and having the inputs and outputs previously described with reference to FIG. 10. An efficient implementation of the AOU cross-bar switch 200 may include multiplexors, tri-state buffers, CLB-based logic, direct wiring, or any subset of such elements joined by reconfigurable couplings. For an outer-loop ISA, the AOU cross-bar switch 200 is preferably implemented to maximize serial address movement in a minimum amount of time, while also providing a maximum number of unique address movement cross-bar couplings to support generalized outer-loop ISA address operate instructions.

Detailed Description Text (57):
Referring now to FIG. 11B, a block diagram of a second exemplary embodiment of the AOU 66 configured for the implementation of an inner-loop ISA is shown. Preferably, an inner-loop ISA requires hardware for performing a very limited set of address operations, and hardware for maintaining at least one source address pointer and a corresponding number of destination address pointers. Types of inner-loop processing for which a very limited number of address operations or even a single address operation are required include block, raster, or serpentine operations upon image data; bit reversal operations; operations upon circular buffer data; and variable length data parsing operations. Herein, a single address operation is considered, namely, an increment operation. Those skilled in the art will recognize that hardware that performs increment operations may also be inherently capable of performing decrement operations, thereby providing an additional address operation capability. In the second exemplary embodiment of the AOU 66, the store/count logic 202 comprises at least one source address register 252 having an input, an output, and a control input; at least one destination address register 254 having an input, an output, and a control input; and a data selector 250 having an input, a control

input, and a number of outputs equal to the total number of source and destination address registers 252, 254 present. Herein, a single source address register 252 and a single destination address register 254 are considered, and hence the data selector 250 has a first output and a second output. The address operate logic 204 comprises a NIPAR 232 having an input, an output, and a control output; and a multiplexor 260 having a number of inputs equal to the number of data selector outputs, a control input, and an output. Herein, the multiplexor 260 has a first input and a second input. The address multiplexor 206 preferably comprises a multiplexor having a number of inputs one greater than the number of data selector outputs, a control input, and an output. Thus, herein the address multiplexor 206 has a first input, a second input, and a third input. The AOU cross-bar switch 200 preferably comprises a conventional cross-bar switch network having bidirectional and unidirectional cross-bar couplings, and having the inputs and outputs previously described with reference to FIG. 10. An efficient implementation of the AOU cross-bar switch 200 may include multiplexors, tri-state buffers, CLB-based logic, direct wiring, or any subset of such elements joined by reconfigurable couplings. For an inner-loop ISA, the AOU cross-bar switch 200 is preferably implemented to maximize parallel address movement in a minimum possible time, while also providing a minimum number of unique address movement cross-bar couplings to support inner-loop address operations.

Detailed Description Text (78):
The GPIM 16 is a conventional interconnect mesh that facilitates point-topoint parallel message routing between interconnect I/O units 304. In the preferred embodiment, the GPIM 16 is a wire-based k-ary n-cube static interconnect network. Referring now to FIG. 16, a block diagram of an exemplary embodiment of a General Purpose Interconnect Matrix 16 is shown. In FIG. 16, the GPIM 16 is a toroidal interconnect mesh, or equivalently, a k-ary 2-cube, comprising a plurality of first communication channels 380 and a plurality of second communication channels 382. Each first communication channel 380 includes a plurality of node connection sites 384, as does each second communication channel 382. Each interconnect I/O unit 304 in the system 10 is preferably coupled to the GPIM 16 such that the message input line 314 and the message output line 316 join consecutive node connection sites 384 within a given communication channel 380, 382. In the preferred embodiment, each T-machine 14 includes an interconnect I/O unit 304 coupled to the first communication channel 380 and an interconnect I/O unit 304 coupled to the second communication channel 382 in the manner described above. The common interface and control unit 302 within the T-machine 14 preferably facilitates the routing of information between its interconnect I/O unit 304 coupled to the first communication channel and its interconnect I/O unit 304 coupled to the second communication channel 382. Thus, for a T-machine 14 having an interconnect I/O unit 304 coupled to the first communication channel labeled as 380c and an interconnect I/O unit 304 coupled to the second communication channel labeled as 382c in FIG. 16, this T-machine's common interface and control unit 302 facilitates information routing between this set of first and second communication channels 380c, 382c.

Detailed Description Text (83):
The system 10 of the present invention is thus particularly useful for problems that can be divided both spatially and temporally into one or more data-parallel subproblems, for example: image processing, medical data processing, calibrated color matching, database computation, document processing, associative search engines, and network servers. For computational problems with a large array of operands, data parallelism exists when algorithms can be applied so as to offer an effective computational speed-up through parallel computing techniques. Data parallel problems possess known complexity, namely, O(n.sup.k). The value of k is problem-dependent; for example, k=2 for image processing, and k=3 for medical data processing. In the present invention, individual S-machines 12 are preferably utilized to exploit data parallelism at the level of program instruction groups. Because the system 10 includes multiple S-machines 12, the system 10 is preferably utilized to exploit data parallelism at the level of sets of entire programs.

Current US Original Classification (1):
712/30

Current US Cross Reference Classification (1):

710/317

Current US Cross Reference Classification (2):
712/20

Current US Cross Reference Classification (3):
712/200

Current US Cross Reference Classification (4):
712/29

Other Reference Publication (19):
E-Mail with Dialog search results.

US005794062A

United States Patent [19]

Baxter

[11] Patent Number: 5,794,062

[45] Date of Patent: Aug. 11, 1998

[54] **SYSTEM AND METHOD FOR DYNAMICALLY RECONFIGURABLE COMPUTING USING A PROCESSING UNIT HAVING CHANGEABLE INTERNAL HARDWARE ORGANIZATION**

[75] Inventor: **Michael A. Baxter**, Sunnyvale, Calif.

[73] Assignees: **Ricoh Company Ltd.**, Tokyo, Japan; **Ricoh Corporation**, Menlo Park, Calif.

[21] Appl. No.: **423,560**

[22] Filed: **Apr. 17, 1995**

[51] **Int. Cl.**[6] **G06F 15/16**

[52] **U.S. Cl.** **395/800.3**; 395/311; 395/800.29; 395/800.2; 395/376

[58] **Field of Search** 395/800, 301, 395/489, 578, 800.29, 800.15, 800.3, 800.28, 307, 284, 311, 312, 200.5, 200.53, 200.54, 800.2, 376; 364/DIG. 2

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,037,094	7/1977	Vandierendonck	364/716
4,250,545	2/1981	Blahut et al.	364/200
4,811,214	3/1989	Nosenchuk et al.	364/200
5,036,473	7/1991	Butts et al.	364/489
5,042,004	8/1991	Agrawal et al.	395/590
5,068,823	11/1991	Robinson	395/500
5,109,353	4/1992	Sample et al.	364/578
5,280,474	1/1994	Nickolis et al.	370/60
5,361,373	11/1994	Gilson	395/800
5,386,562	1/1995	Jain et al.	395/650
5,430,734	7/1995	Gilson	371/22.2
5,452,101	9/1995	Keith	358/426
5,457,408	10/1995	Leung	326/38
5,465,975	11/1995	Thepaut et al.	395/800

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

9410627	11/1994	WIPO .
9509392	6/1995	WIPO .

OTHER PUBLICATIONS

Riley, David D. and Baron, Robert J., Design and Evaluation of a Synchronous Triangular Interconnection Scheme for Interprocessor Communications, IEEE Transactions On Computers, vol. C-31, No. 2, Feb. 1982, pp. 110–118.

(List continued on next page.)

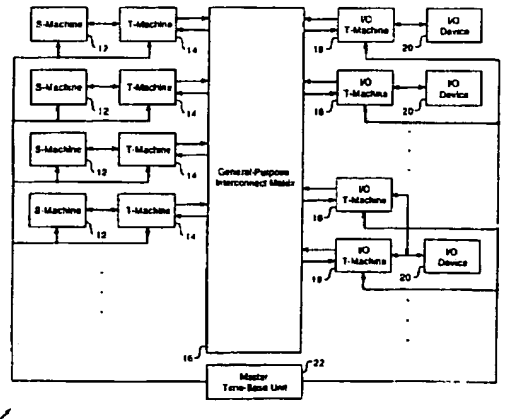
Primary Examiner—Meng-Ai T. An

Attorney, Agent, or Firm—Fenwick & West, LLP

[57] **ABSTRACT**

A set of S-machines, a T-machine corresponding to each S-machine, a General Purpose Interconnect Matrix (GPIM), a set of I/O T-machines, a set of I/O devices, and a master time-base unit form a system for scalable, parallel, dynamically reconfigurable computing. Each S-machine is a dynamically reconfigurable computer having a memory, a first local time-base unit, and a Dynamically Reconfigurable Processing Unit (DRPU). The DRPU is implemented using a reprogrammable logic device configured as an Instruction Fetch Unit (IFU), a Data Operate Unit (DOU), and an Address Operate Unit (AOU), each of which are selectively reconfigured during program execution in response to a reconfiguration interrupt or the selection of a reconfiguration directive embedded within a set of program instructions. Each reconfiguration interrupt and each reconfiguration directive references a configuration data set specifying a DRPU hardware organization optimized for the implementation of a particular Instruction Set Architecture (ISA). The IFU directs reconfiguration operations, instruction fetch and decode operations, memory access operations, and issues control signals to the DOU and the AOU to facilitate instruction execution. The DOU performs data computations, and the AOU performs address computations. Each T-machine is a data transfer device having a common interface and control unit, one or more interconnect I/O units, and a second local time-base unit. The GPIM is a scalable interconnect network that facilitates parallel communication between T-machines. The set of T-machines and the GPIM facilitate parallel communication between S-machines.

22 Claims, 23 Drawing Sheets



U.S. PATENT DOCUMENTS

5,466,117	11/1995	Resler et al.	414/799
5,475,624	12/1995	West	364/578
5,475,856	12/1995	Kogge	395/800
5,497,498	3/1996	Taylor	395/800
5,511,173	4/1996	Yamaura et al.	395/598
5,522,083	5/1996	Gove et al.	395/800
5,524,243	6/1996	Gheorghiu	395/651
5,535,342	7/1996	Taylor	395/307
5,535,406	7/1996	Kolchinsky	395/800
5,539,888	7/1996	Byers et al.	395/581
5,539,893	7/1996	Thompson et al.	395/449
5,542,067	7/1996	Chappell et al.	395/494
5,546,347	8/1996	Ko et al.	365/221
5,546,545	8/1996	Rich	395/291
5,546,562	8/1996	Patel	395/500
5,548,771	8/1996	Davis et al.	395/800
5,548,775	8/1996	Hershey	395/800
5,550,845	8/1996	Traub	371/27
5,550,989	8/1996	Santos	395/306
5,551,013	8/1996	Beausoleil	395/500
5,557,734	9/1996	Wilson	395/162
5,600,845	2/1997	Gilson	395/800

OTHER PUBLICATIONS

Casselman, Steven, Virtual Computing and the Virtual Computer, Virtual Computer Corporation, 1993 IEEE, pp. 43–48.

"XCELL", The Quarterly Journal for Xilinx Programmable Logic Users, Issue 16, First Quarter 1995, pp. 23–29.

Chapman, Ken, "Dynamic Microcontroller Using XC4000 FPGAs", XILINX, Dec. 1994, pp. 1–15.

Xilinx Application Notes, *Trademarks and Patents*, 1995.

Alfke, Peter and New, Bernie, "Quadrature Phase Decoder," XILINX Application Note (XAPP 012.001), pp. 8–173 to 8–174.

New, Bernie, "Estimating the Performance of XC4000 Adders and Counters," XILINX Application Note (XAPP 018.00), pp. 8.116–8.118.

Weiss, Ray, "Viva la revolucion!" *Computer Design*, Sep. 1995, p. 109.

Baker, Stan, "Quo vadis, EDA?" *Electronic Engineering Times*, Jun. 26, 1995, p. 83.

Wirbel, Loring, "Dolphin's KSR buy hikes SCI outlook," *Electronic Engineering Times*, Nov. 13, 1995, pp. 37, 42.

Forrest, John, "Software Acceleration Architectures," Software Acceleration Internet Home Page, Sep. 1995.

Razdan, Rahul, "PRISC: Programmable Reduced Instruction Set Computers," Doctoral Thesis (Division of Applied Sciences: Computer Science), Harvard University, May 1994.

Razdan, Rahul and Smith, Micahel D, "A High–Performance Microarchitecture with Hardware–Programmable Functional Units," Digital Equipment Corporation (Hudson, MA), 1994, pp. 1–9.

Athanas, Peter, "The Hokie Instant RISC Microprocessor," Electrical Engineering, VISC Internet Home Page, 1996.

Depreitere, J., Neefs, H., Van Marck, H. and Van Campenhout, J. "A Hybrid Optoelectronic 3–D Field Programmable Gate Array Demonstrator," University of Ghent, Dept. of Electronics and Information Systems (Belgium).

Wirthlin, Michael J. and Hutchings, Brad L., "A Dynamic Instruction Set Computer,"Dept. of Electrical and Computer Eng. (Brigham Young University).

Wirthlin, Michael J., Huchings, Brad L. and Gilson, Kent L. "The Nano Processor: a Low Resource Reconfigurable Processor," *IEEE*, 1994, pp. 23–30.

Dahl, Matthew et al. "Emulation of the Sparcle Microprocessor with the MIT Virtual Wires Emulation System," MIT Laboratory for Computer Science (1994 IEEE), pp. 14–22.

E–Mail with Dialog search results.

ATLightspeed promotional literature including a 3D Scanline Texture Mapping Algorithm.

GO Giga Operations Data Sheet for "Reconfigurable Interface Card™ G900 RIC™", G900 RIC.

GO Giga Operations Data Sheet for "Reconfigurable Computing (RC) Developmental Software".

GO Giga Operations—"FCCM 1995 & FCCM 1996 follow up:".

GO Giga Operations Data Sheet for "X213EMOD", Release 6, Apr. 1, 1996.

GO Giga Operations Web page on "Intellectual Property" 1995.

CMP Publications via Fulfillment by individual, Inc.; Jul 30, 1996.

Abstract from "Video Processing Module Using a Second Programmable Logic Device Which Reconfigures a First Programmable Logic Device for Data Transformation" (Giga Operations Corporation).

Hadley, James D., ande Hutchings Brad L., "Design Methodologies for Partially Reconfigured Sysems" *Dept. of Electrical and Computer Eng.*

Galloway, David, "The Transmogrifier C Hardware Description Language and Compiler for FPGAs" *Department of Electrical and Computer Engineering* (Date Unknown).

Jones, Chris et al., "Issues in Wireless Video Coding using Run–time–reconfigurable FPGAs" *Electrical Engineering Department* (Date Unknown).

Lemoine, Eric and Merceron David, "Run Time Reconfiguration of FPGA for Scanning Genomic Databases" (Date Unknown).

"Advanced Features Squeeze onto Processor Chip", Jim Slager, Computer Design, Oct. 1983, pp. 189–193.

L9: Entry 2 of 2 File: USPT Feb 24, 1998

DOCUMENT-IDENTIFIER: US 5721922 A
TITLE: Embedding a real-time multi-tasking kernel in a non-real-time operating system

Brief Summary Text (8):
In many graphic user interface environments, e.g. Windows.RTM. (Windows.RTM. is a registered trademark of Microsoft Corporation of Redmond, Wash.), if audio or video playback alone is being executed, a system processing the natural data types will produce a smooth playback output. However, other operations being performed on a system processing the natural data types, such as spreadsheet and networking activities, as well as transmission of electronic mail (E-mail), can disrupt the scheduling of the audio and video playback. Such disruptions may cause the audio and video playback to manifest pops, clicks and/or jerky video output. In addition, playback of both audio as well as video data may produce an unnatural output due to lack of synchronization of the natural data types.

Current US Cross Reference Classification (3):
710/20

Current US Cross Reference Classification (4):
710/36

Current US Cross Reference Classification (5):
710/48

Current US Cross Reference Classification (6):
710/52

Current US Cross Reference Classification (7):
710/6

CLAIMS:

13. The apparatus of claim 10 further comprising a real-time interrupt handler for calling said real-time task, enabling a scheduling lock to prevent said real-time scheduler from switching tasks, scheduling said real-time event with said virtual machine manager to process said real-time tasks in said V.times.D event mode, making said real-time tasks ready for execution, and processing rescheduling to cause all ready real-time tasks to be executed over lower priority idle task.

27. The system of claim 24 further comprising a real-time interrupt handler for calling said real-time task, enabling a scheduling lock to prevent said real-time scheduler from switching tasks, scheduling said real-time event with a virtual machine manager to process said real-time tasks in said V.times.D event mode, making said real-time tasks ready for execution, and processing rescheduling to cause all ready real-time tasks to be executed over lower priority idle task.

US005721922A

United States Patent [19]

Dingwall

[11] Patent Number: 5,721,922

[45] Date of Patent: Feb. 24, 1998

[54] **EMBEDDING A REAL-TIME MULTI-TASKING KERNEL IN A NON-REAL-TIME OPERATING SYSTEM**

[75] Inventor: **Thomas J. Dingwall**, Portland, Oreg.

[73] Assignee: **Intel Corporation**, Santa Clara, Calif.

[21] Appl. No.: **323,044**

[22] Filed: **Oct. 13, 1994**

[51] **Int. Cl.**[6] **G06F 9/46**

[52] **U.S. Cl.** **395/673**; 395/677; 395/572; 395/806; 395/826; 395/840; 395/856; 395/868

[58] **Field of Search** 395/650, 673, 395/677, 872, 806, 826, 840, 856, 868; 364/280, 281.3

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,081,577	1/1992	Hatle	395/827
5,414,848	5/1995	Sandage et al.	395/650
5,469,571	11/1995	Bunnell	395/700
5,515,538	5/1996	Kleiman	395/733
5,528,513	6/1996	Vaitzblit et al.	364/514 A

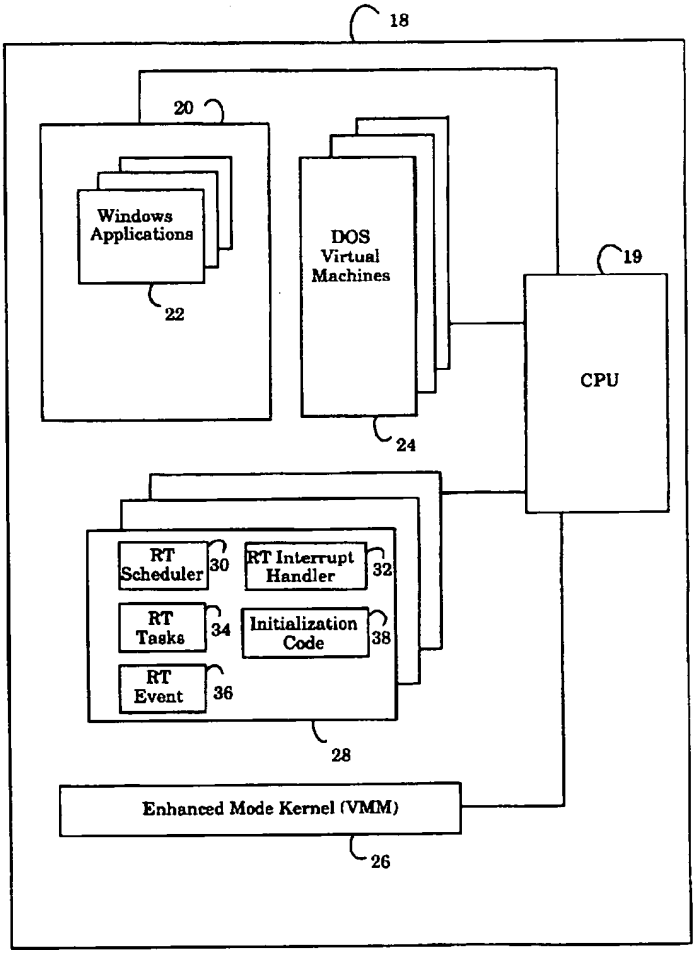
Primary Examiner—Lucien U. Toplu

Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

The invention provides a method and apparatus for embedding a real-time multi-tasking kernel in a non-real-time operating system. Through encapsulating a real-time kernel into the interrupt handling environment of a non-real-time graphical user interface, such as Windows®, the method of the present invention allows for an entire real-time environment to be supported within the graphical user interface. The scheduler of the real-time kernel supports multiple threads of execution all running at higher priority than the graphical user interface tasks. By using synchronization mechanisms of the graphical user interface, e.g. VxD events in enhanced mode Windows®, the real-time tasks are able to make use of system services of the graphical user interface.

36 Claims, 3 Drawing Sheets



WEST

Generate Collection

L12: Entry 2 of 6 File: USPT Apr 2, 2002

DOCUMENT-IDENTIFIER: US 6366951 B1
TITLE: Distributed processing system where a management computer automatically connects remote reduced-capability workstations with centralized computing modules

Detailed Description Text (24):
The management computer 640 loads a comprehensive suite of diagnostic routines in each computer module 610 not currently connected to a workstation 100 using the dedicated management computer network 635. The results of the diagnostic routines are monitored and logged by the management computer 640 using an interrupt driven technique across the same management computer network 635. In the event a computer module 610 fails the diagnostic routines, the management computer 640 automatically locks out that computer module 610, thereby preventing its use by any workstation 100, and notifying the system administrator of the failure via e-mail or other means.

Current US Cross Reference Classification (3):
712/31

US006366951B1

(12) **United States Patent**
Schmidt

(10) **Patent No.: US 6,366,951 B1**
(45) **Date of Patent: *Apr. 2, 2002**

(54) **DISTRIBUTED PROCESSING SYSTEM WHERE A MANAGEMENT COMPUTER AUTOMATICALLY CONNECTS REMOTE REDUCED-CAPABILITY WORKSTATIONS WITH CENTRALIZED COMPUTING MODULES**

(76) Inventor: **Curt A. Schmidt**, 16178 E. Sierra Pass Way, Hacienda Heights, CA (US) 91745

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/017,398**

(22) Filed: **Feb. 2, 1998**

Related U.S. Application Data

(60) Provisional application No. 60/037,482, filed on Feb. 3, 1997, and provisional application No. 60/037,481, filed on Feb. 3, 1997.

(51) **Int. Cl.**[7] **G06F 15/16**; G06F 15/177

(52) **U.S. Cl.** **709/208**; 709/220; 709/223; 712/31

(58) **Field of Search** 709/208, 220, 709/223, 201; 700/3; 710/110; 712/31; 345/520, 522, 2.1

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,737,950	A	* 4/1988	Fechalos	370/56
5,315,711	A	* 5/1994	Barone et al.	340/706
5,337,229	A	* 8/1994	Holland	700/2
5,437,014	A	* 7/1995	Busboom et al.	340/706
5,499,377	A	* 3/1996	Lee	709/244
5,577,205	A	* 11/1996	Hwang et al.	709/208

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0 359 064	3/1990
WO	WO 94/22088	9/1994

OTHER PUBLICATIONS

Halsall, F., "Data Communications, Computer Networks, and Open Systems," 4th ed., Addison–Wesley, pp. 355–357, 1996.*

Edwards, B., "WinFrame Works Wonders," LAN Times, vol. 13, No. 24, p. 87, Oct. 1996.*

Henderson, T., "Center of the Universe," Windows Magazine, vol. 7, No. 10, pp. 244, Oct. 1996.*

Stanczak, M., "Not Just a 'Dumb' Terminal: Wyse Win Term Model 2000 Units Edge into Internet Terminal Area," PC Week, vol. 13, No. 5, p. 59, Feb. 1996.*

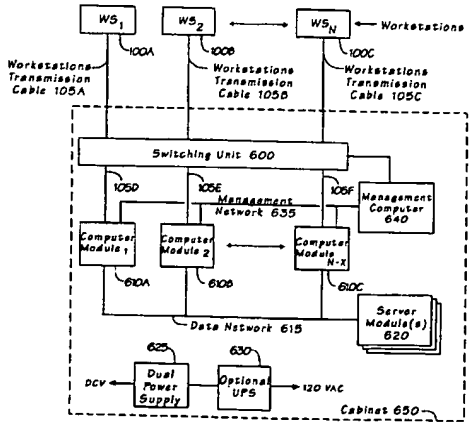
Flanagan, W., "ICA greets the NC," Computer Shopper, vol. 16, No. 12, p. 624, Dec. 1996.*

Primary Examiner—Zarni Maung
Assistant Examiner—Andrew T. Caldwell
(74) *Attorney, Agent, or Firm*—Christie, Parker & Hale, LLP

(57) **ABSTRACT**

A distributed computer system having centrally available processing units. A plurality of computer workstations are connected to a plurality of processing units by computer transmission cables. In one embodiment each of the workstations may be connected to any of the plurality of processing units. These connections are accomplished by a switching unit controlled by a management computer where the management computer automatically connects and disconnects individual computer workstations to individual processing units. In a typical embodiment, there are more computer workstations then processing units and the management computer disconnects idle computer workstations from processing units and connects previously idle computer workstations that become active to processing units not then connected to another computer workstation. A portion of a video display adapter is included in computer workstations while a second portion of a video display adapter is included in the processing units.

3 Claims, 11 Drawing Sheets



Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L12	L11 and ((attribut$ or message$) with (contain$ or includ$ or identif$) with (handler$ or program))	63	L12
L11	l1 and (e-mail).ab.	219	L11
L10	L1 and (attribut$ with (includ$ or contain$) with (identif$) with (handler$))	11	L10
L9	L1 and (message$ with (includ$ or contain$) with (identif$) with (handler$))	19	L9
L8	L7 and (e-mail or email)	8	L8
L7	L3 and (attribut$ with identif$ with handler$)	8	L7
L6	L3 and (message$ with identif$ with handler$)	22	L6
L5	L3 and (identif$ with handler$)	40	L5
L4	L3 and (identif$ with hanfler$)	0	L4
L3	L1 and (message$ with (flag$ or screening))	899	L3
L2	L1 and (message$ with (flag$ or screening))	899	L2
L1	((709/$)!.CCLS.)	15581	L1

END OF SEARCH HISTORY

L9: Entry 6 of 19 File: USPT Jul 17, 2001

DOCUMENT-IDENTIFIER: US 6263360 B1
TITLE: System uses filter tree and feed handler for updating objects in a client from a server object list

Brief Summary Text (18):
According to yet another aspect of the present invention, a method for updating a first object on a client that is arranged to be a part of a client/server object-based computing environment includes determining whether there is a second object that is referenced by the first object when the first object is in existence on the client, and sending an update message to a handler that is associated with the first object when the first object is in existence on the client. The method also includes adding a reference that identifies the second object to the first object when it is determined that the second object is referenced by the first object, and sending a notification message to the handler that is arranged to indicate the existence of the second object when it is determined that the second object is referenced by the first object.

Detailed Description Text (70):
After acknowledgment of the receipt of the message is sent to the server, a determination is made in step 1402 as to whether the message is a message to update an object on the client. In general, a message to update an object on the client may include such information as the name of the object, identifiers pertaining to other objects which may reference the object that is to be updated, and a class name for the handler class associated with the object. When it is determined that the message is an object update message, then process flow moves to step 1403 in which the object update message is handled. The steps associated with handling the object update message will be described below with reference to FIG. 15. Once the object update message is handled, the steps associated with listening for updates is allowed to continue. In other words, the client continues to listen for updates.

Current US Original Classification (1):
709/203

US006263360B1

(12) **United States Patent**
Arnold et al.

(10) **Patent No.: US 6,263,360 B1**
(45) **Date of Patent: Jul. 17, 2001**

(54) **SYSTEM USES FILTER TREE AND FEED HANDLER FOR UPDATING OBJECTS IN A CLIENT FROM A SERVER OBJECT LIST**

(75) Inventors: **James F. Arnold**, Helena, MT (US); **Scott W. Shaw**, San Francisco, CA (US); **Nathan W. Williams**, Helena, MT (US); **D. Scott Seaton**, Fremont, CA (US); **Carla P. Woodworth**, San Mateo, CA (US); **Jeffrey Casper**, Mountain View, CA (US)

(73) Assignee: **SRI International**, Menlo Park, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/087,799**

(22) Filed: **Jun. 1, 1998**

(51) **Int. Cl.**[7] **G06F 13/38**; G06F 15/17

(52) **U.S. Cl.** **709/203**; 707/10

(58) **Field of Search** 709/203, 303, 709/304, 204; 707/10, 103; 705/5, 10

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,339,392	*	8/1994	Risberg et al.	345/333
5,754,771		5/1998	Epperson et al.	395/200
5,812,844	*	9/1998	Jones et al.	709/104
5,813,007	*	9/1998	Nielsen	707/10
5,826,270	*	10/1998	Rutkowski et al.	707/10
5,835,087	*	11/1998	Herz et al.	345/327
5,835,911	*	11/1998	Nakagawa et al.	707/203
5,918,013	*	6/1999	Mighdoll et al.	709/217
5,918,214	*	6/1999	Perkoski	705/27
5,920,725	*	7/1999	Ma et al.	395/712
5,937,189	*	8/1999	Branson et al.	395/701
5,948,066	*	9/1999	Whalen et al.	709/229
5,959,543	*	9/1999	Laporta et al.	340/825.44
5,978,577	*	11/1999	Rierden et al.	709/203
6,029,175	*	2/2000	Chow	707/104
6,041,360	*	3/2000	Himmel et al.	709/245

FOREIGN PATENT DOCUMENTS

0553560A2	8/1993	(EP) .
0817031A2	1/1998	(EP) .
WO98/02812	1/1998	(WO) .
WO98/19239	5/1998	(WO) .

OTHER PUBLICATIONS

Feiler et al., Propagator: a family of patterns, 12 p, Aug. 1997.*

Canos et al., KAOS: an object–oriented software tool for the objects definition updating, querying and programming in an object–oriented environment, 5 p, May 1998.*

Plasil et al, SOFA/DCUP: architecture for component trading and dynamic updating, 10 p, May 1998.*

Bukhres et al., Mobile Computing in Military Ambulatory Care, 1997, Proceedings from the Tenth IEEE Symposium on Computer–Based Medical Systems.

IBM Technical Disclosure Bulletin, Event Notification Mechanism, 1993, vol. 36, No. 4.

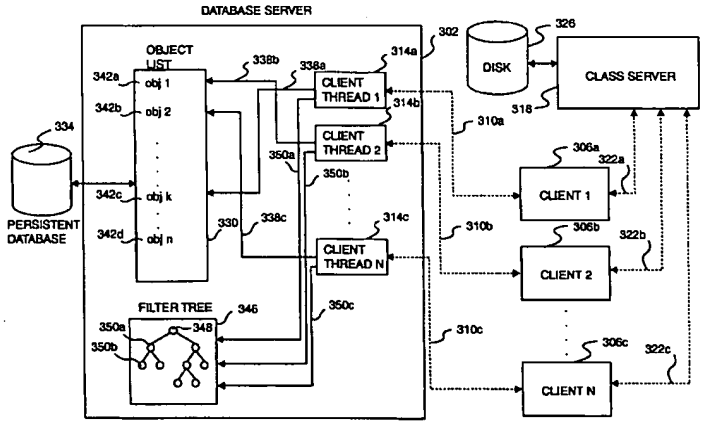
* cited by examiner

Primary Examiner—Le Hien Luu
Assistant Examiner—Bunjob Jaroenchonwanit
(74) *Attorney, Agent, or Firm*—Beyer Weaver & Thomas, LLP

(57) **ABSTRACT**

Methods and apparatus for maintaining updated information on client/server object-oriented computing systems are disclosed. In accordance with one aspect of the present invention, a client/server object-based computing system includes a server which has a server object list that is arranged to include objects that are to be updated. The system also includes a client which has a client object list that contains an object in which the client has interest. The objects that are to be updated include the object in which the client has interest, and the server is arranged to send a message to the client that indicates that the object in which the client has interested should be updated on the client. In one embodiment, the client and the server are in communication over a low-bandwidth link.

22 Claims, 22 Drawing Sheets



L9: Entry 7 of 19 File: USPT May 16, 2000

DOCUMENT-IDENTIFIER: US 6065123 A
**** See image for Certificate of Correction ****
TITLE: Computer system with unattended on-demand availability

Detailed Description Text (75):
The IonConnect procedure call is utilized by the calling application process to connect to InstantON servicing agent 140 and declare its readiness to receive InstantON messages from other applications. An application process which is not connected will not receive any InstantON messages from other applications, even if it is currently running. When calling IonConnect, the application process includes as parameters the registration identifier of the application process, and a message handler and message identifier for receiving InstantON messages scheduled by IonScheduleAction, discussed in more detail below. Upon receiving the IonConnect call, InstantON servicing agent 140 associates the registration identifier supplied by the call with the calling application process. In one implementation, this association is made by storing the operating system process identification in process identifier field 430 of registration record 400. In one implementation, InstantON servicing agent 140 returns a value indicating success, an invalid registration identifier, or that InstantON servicing agent 140 is unavailable.

Current US Cross Reference Classification (1):
709/103

US006065123A

United States Patent [19]

Chou et al.

[11] Patent Number: 6,065,123

[45] Date of Patent: May 16, 2000

[54] **COMPUTER SYSTEM WITH UNATTENDED ON-DEMAND AVAILABILITY**

[75] Inventors: **Stephen T. Chou**, Beaverton; **Russell J. Fenger**, Aloha; **Mohan J. Kumar**; **Victor B. Lortz**, both of Beaverton; **Benjamin L. Manny**, Portland; **Mil Travnicek**, Portland; **Chih-Kan Wang**, Portland, all of Oreg.

[73] Assignee: **Intel Corporation**, Santa Clara, Calif.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/978,545**

[22] Filed: **Nov. 26, 1997**

Related U.S. Application Data

[60] Continuation of application No. 08/726,694, Oct. 7, 1996, abandoned, which is a division of application No. 08/400,027, Mar. 6, 1995, abandoned.

[51] **Int. Cl.**[7] **G06F 1/32**; G06F 9/00

[52] **U.S. Cl.** **713/322**; 713/324; 709/3

[58] **Field of Search** 395/750.01–750.06, 395/670–678; 364/707

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,203,153	5/1980	Boyd	713/323
4,259,594	3/1981	Fox et al.	307/141
4,284,849	8/1981	Anderson et al.	379/38
4,317,180	2/1982	Lies	364/707
4,521,847	6/1985	Ziehm et al.	364/184
4,648,031	3/1987	Jenner	395/182.08
4,660,168	4/1987	Grant et al.	395/208
4,800,521	1/1989	Carter et al.	395/672
5,151,987	9/1992	Abraham et al.	395/182.18
5,363,505	11/1994	Maslak et al.	395/670
5,375,247	12/1994	Hueser	395/750
5,423,045	6/1995	Kannan et al.	395/750
5,426,775	6/1995	Boccon-Gibod	714/36
5,452,401	9/1995	Lin	713/322
5,495,617	2/1996	Yamada	713/323
5,511,205	4/1996	Kannan et al.	395/750
5,519,874	5/1996	Yamagishi et al.	395/800
5,524,241	6/1996	Ghoneimy et al.	395/610
5,530,878	6/1996	Bauer et al.	395/750
5,530,879	6/1996	Crump et al.	395/750
5,560,022	9/1996	Dunstan et al.	395/750
5,586,332	12/1996	Jain et al.	713/322
5,592,173	1/1997	Lau et al.	342/357
5,634,131	5/1997	Matter et al.	713/322
5,666,537	9/1997	Debnath et al.	713/322
5,692,204	11/1997	Rawson et al.	713/340

OTHER PUBLICATIONS

Supplementary Partial European Search Report for EPO Application No. EP 96 90 8571 dated Jan. 9, 1998.

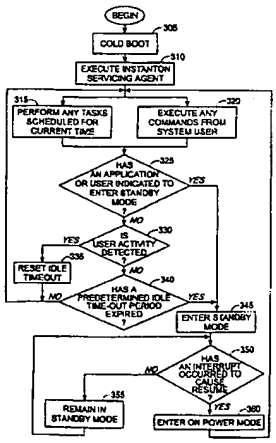
Patent Abstracts of Japan Application No. 58008162 dated Jan. 21, 1983, entitled "Automatic Operating System for Power Restoration", Inventor Murata Shuichi.

Primary Examiner—Ayaz R. Sheikh
Assistant Examiner—Sumati Lefkowitz
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman LLP

[57] **ABSTRACT**

A computer system with unattended on-demand availability includes power-saving features which place the system into a Standby mode whenever the system is idle or is not being used. Prior to entering Standby mode, the system sets a hardware timer which indicates when the next scheduled event in the system should be performed. When either the timer expires or another event occurs which requires system operation, the system resumes to the On power state without user intervention. In one embodiment, the system of the present invention allows applications to periodically save their operational states. By saving their operational states, applications are able to guard against power failures and crashes. If a power failure or crash occurs, the system consults restart policies and, if appropriate, automatically re-starts applications to their most recently saved operational states once power is re-stored.

20 Claims, 10 Drawing Sheets



L9: Entry 1 of 10 File: USPT Oct 20, 1998

DOCUMENT-IDENTIFIER: US 5826101 A
TITLE: Data processing device having split-mode DMA channel

Brief Summary Text (19):
The increasing demands of technology and the marketplace make desirable even further structural and process improvements in processing devices, systems and methods of operation. These demands have lead to increasing the performance of single-chip devices and single systems as state-of-the-art silicon processing technologies allow. However, some performance-hungry applications such as video conferencing, 3D graphics and neural networks require performance levels over and above that which can be achieved with a single device or system. Many such applications benefit from parallel processing.

Detailed Description Text (79):
For example, shown in FIG. 7a is trap address logic 208 containing trap vector table pointer register 207, adder 209, program counter +4 (PC+4) register 210. During system cycle clock cycle 620 (after fetching the LAT instruction, control logic 202 decodes the LAT instruction. Trap number (TN) which specifies a particular trap routine is extracted from the LAT instruction by decoder 202a and combined with trap vector table pointer (TVTP) register 207 using adder 209. The result is a trap address (TA) specifying a location in memory that contains the trap vector which is the address of the first instruction for the trap routine to be executed. The contents of the TVTP register 207 can be altered thus offering even more flexibility in placing trap routines within the memory map of microcomputer 10. During the third cycle of the system clock after fetching the LAT instruction, the trap address is sent to memory via bus 30a to access the trap vector that is received on bus 30d. Access to memory is in accordance to above herein described technique. On the fourth cycle of the system clock, the current contents of program counter register 92 is transferred to PC+4 register 210 and the trap vector is transferred to program counter 92. Thus, program counter register 92 contains the first instruction of the trap routine, and the previous contents of the program counter register 92 are stored in PC+4 register 210. When the trap routine is complete, the contents of PC+4 are transferred back to program counter register 92 and program execution resumes at the point where the trap routine interrupted. Advantageously, the trap routine interrupts program execution using only one system cycle clock cycle and continues to take advantage of the pipelining scheme by keeping the pipeline full while indirection of program execution is occurring.

Detailed Description Text (224):
FIG. 19 shows an embodiment of a stand alone configuration of the improved data processing configured to show connections to a plurality of memories 350 and 351 and peripheral devices 360 and 361. Global peripheral port 24 and local peripheral port 26 provide the interface to the external devices. For example, bus 380 can be used for program accesses and bus 390 can be used for data or I/O accesses which allows for simultaneous external program and data accesses. Microcomputer 10 also has available six communication channels capable of interfacing to other systems in I/O intensive applications. Peripherals and other external devices such as key boards, monitors, disk drives, printers, displays, transducers, modems, processors, local area networks (LANs), and other known or hereafter devised with which the system commends its use can be connected to the peripheral ports 24 and 26 and communication ports 50-55.

Detailed Description Text (235):
Applications that utilize complex algorithms are well suited for the herein-described preferred embodiments. Such applications include speech-recognition

technology, cellular radio phones, video teleconferencing, and multiplexing four voice conversations on leased 64-Kbit/s lines that formerly could carry only one. A large number of other computationally-intensive problems are well-suited for parallel processing, such as 3D graphics, control, array processors, neural networks, and numerous other applications listed in the coassigned applications incorporated herein by reference.

Current US Original Classification (1):
712/34

Current US Cross Reference Classification (1):
710/22

US005826101A

United States Patent [19]

Beck et al.

[11] Patent Number: 5,826,101

[45] Date of Patent: Oct. 20, 1998

[54] **DATA PROCESSING DEVICE HAVING SPLIT-MODE DMA CHANNEL**

[75] Inventors: **Michael D. Beck**, Houston; **Laurence R. Simar**, Richmond, both of Tex.

[73] Assignee: **Texas Instruments Incorporated**, Dallas, Tex.

[21] Appl. No.: **109,201**

[22] Filed: **Aug. 19, 1993**

Related U.S. Application Data

[63] Continuation of Ser. No. 590,106, Sep. 28, 1990, abandoned.

[51] **Int. Cl.**[6] **G06F 13/00**

[52] **U.S. Cl.** **395/800.34**; 395/842

[58] **Field of Search** 395/800, 725, 395/425, 400, 800.01, 800.43, 200.42, 200.43, 200.68, 840, 842, 856, 800.34; 365/189.01, 189.04, 189.05, 220, 230.01, 230.02, 230.05; 711/100, 111, 200, 202, 211; 340/825, 825.03

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,439,839	3/1984	Kneib et al.	364/900
4,577,282	3/1986	Caudel et al.	364/200
4,646,232	2/1987	Chang et al.	364/200
4,658,350	4/1987	Eggebrecht et al.	395/425
4,713,748	12/1987	Magar et al.	364/200
4,805,094	2/1989	Oye et al.	395/250
4,829,475	5/1989	Ward et al.	365/78
4,833,649	5/1989	Greub	365/189.06
4,878,190	10/1989	Darley et al.	364/752
4,893,302	1/1990	Hemmady et al.	370/60
4,912,636	3/1990	Magar et al.	364/200
4,920,480	4/1990	Murakami et al.	395/800
4,933,840	6/1990	Sera et al.	364/200
4,959,782	9/1990	Tulpule et al.	364/200
4,980,828	12/1990	Kapcio et al.	364/413.13
5,001,624	3/1991	Hoffman et al.	395/375
5,005,121	4/1991	Nakada et al.	395/800
5,014,247	5/1991	Albachten, III et al.	365/230.05
5,056,010	10/1991	Huang	395/425
5,072,420	12/1991	Conley et al.	395/425
5,099,417	3/1992	Magar et al.	395/425
5,119,487	6/1992	Taniai et al.	395/425
5,151,999	9/1992	Marzucco et al.	395/800
5,163,132	11/1992	DuLac et al.	395/275
5,179,662	1/1993	Corrigan et al.	395/250
5,193,169	3/1993	Ishikawa	395/425

OTHER PUBLICATIONS

Second Generation TMS320 User's Guide, Texas Instruments, Copyright 1987, pp. 3.20, 3.21, 4.2, 4.3, 4.86 and 4.87.

Intelligent Buffer Reconciles Fast Processors and Slow Peripherals; Electronics, Sep. 11, 1980, Daniel L. Hillman, Zilog Inc. Cupertino, California; pp. 131–135.

Primary Examiner—Alpesh M. Shah

Attorney, Agent, or Firm—Scott B. Stahl; James C. Kesterson; Richard L. Donaldson

[57] **ABSTRACT**

A data processing device comprising comprising a memory having a plurality of addressable memory locations, a processor circuit, an input register operative to hold input data, an output register operative to hold output data, and a direct memory access (DMA) circuit operative to receive input data from the input register for storing the input data in a first memory location and to concurrently send output data from a second memory location to said output register. Other devices, systems and methods are also disclosed.

15 Claims, 44 Drawing Sheets